User's Manual
Model X-PORT-10
Revision 0.1

An Intelligent CAMAC-to-FERA Bridge

JTEC Instruments
32 Thompson Rd.
Rochester
Tel: (585)-334-1960; FAX: (585)-334-1960
http://www.jtec-instruments.com/

Information furnished by JTEC Instruments (JTEC) is believed to be accurate and reliable. However, no responsibility is assumed by JTEC for its use, nor for any infringements of patents or other rights of third parties, which may result from its use. No license is granted by implication or otherwise under any patent or patents rights of JTEC. JTEC reserves the right to change specifications at any time without notice.

Copyright 2002 by JTEC

Manual #: mdo-X-PORT-MAN-001.1
February 2002

TABLE OF CONTENTS

1. OVERVIEW ... 3
1.1. FEATURES ... 3
2. SPECIFICATIONS ... 4
2.1. CAMAC PROTOCOL ... 4
2.1.1. CONTROL BUS ... 4
2.1.2. CAMAC DATA BUS USAGE ... 5
2.1.3. CAMAC AUXILIARY BUS ... 5
2.1.4. AUXILIARY BUS ARBITRATION ... 5
2.1.5. CAMAC COMMANDS ... 6
2.2. FERA PROTOCOL ... 6
2.2.1. ECL COMMAND BUS ... 6
2.2.2. ECL PORT ENABLE/PASS ... 6
2.2.3. ECL PORT OUTPUT ... 7
2.3. VARIOUS ECL CONTROL PORT SIGNALS ... 7
2.4. INTERNAL DATA STORAGE FIFO ... 7
2.5. POWER REQUIREMENTS ... 8
3. FUNCTIONAL DESCRIPTION ... 9
3.1. FIRMWARE VERSIONS ... 9
3.1.1. THE STANDARD VERSION ... 9
3.1.2. THE DPP VERSION ... 10
3.2. CAMAC MODULE TYPES AND INDIVIDUAL MODES OF OPERATION ... 10
3.2.1. USER-DEFINED MODULES ... 11
3.2.2. INDIVIDUAL MODULE MODES OF OPERATION ... 11
3.3. GLOBAL MODES OF OPERATION ... 12
3.3.1. TRIGGER MODES AND TIMER SETTINGS ... 12
3.3.2. CAMAC MODULE CLEARING OPTIONS ... 13
3.3.3. Virtual Station Number ... 13
4. OPERATING INSTRUCTIONS ... 13
4.1. HARDWARE SETUP ... 13
4.1.1. ECL PORT PULL-DOWN AND IMPEDANCE MATCHING RESISTORS ... 14
4.1.2. FIRMWARE VERSION SELECTION ... 14
4.1.3. FIFO SETUP ... 14
4.1.4. CONNECTING TO CAMAC BUS ... 15
4.1.5. AUXILIARY PORT CONNECTOR ... 15
4.1.6. AUXILIARY CONTROLLER ARBITRATION ... 15
4.1.7. BOOTING AND REBOOTING ... 16
5. PROGRAMMING OF X-Port10 ... 17
5.1.1. NF(16)A(0)W(VSN) – VIRTUAL STATION NUMBER ... 17
5.1.2. NF(16)A(1)W(NumberOfModules) - NUMBER OF CAMAC MODULES ... 17
5.1.3. NF(16)A(2)W(ModuleData) - CAMAC MODULE DESCRIPTION ... 18
5.1.4. NF(16)A(3 – 6)W(DelaySubtractor) – “FIFO WRITE” DELAY ... 19
5.1.5. NF(16)A(7)W(GlobalMode) - GLOBAL MODE OF OPERATION ... 19
5.1.6. NF(16)A(8)W(Delays) – TRIGGER AND LAM TIMING ... 20
5.1.7. NF(16)A(9, 10)W(LAMMaskX) – LAM MASK ... 21
5.1.8. NF(16)A(11)W(FACodes) – DEFINE A NEW MODULE TYPE ... 21
5.2. NF(0)A(X) – READ-BACK OF THE SETUP DATA ... 22
5.2.1. READ-BACK OF OPERATING PARAMETERS ... 22
5.2.2. NF(0)A(12) - FIRMWARE ID ... 22
5.3. EXAMPLES OF X-Port10 PROGRAMMING ... 23
5.3.1. EXAMPLE 1 – STANDARD VERSION ... 23
5.3.2. EXAMPLE 2 – STANDARD VERSION ... 23
5.3.3. EXAMPLE 3 – DPP VERSION ... 24
6. STRUCTURE OF THE OUTPUT DATA STREAM ... 25
APPENDIX A: SUMMARY OF CAMAC COMMANDS ... 27

APPENDIX B: OPERATIONS INVOLVING THE ATMEL AT29C010A PEROM 28
B.1. LOADING THE FPGA CONFIGURATION DATA INTO AT29C010A 28
B.2. SOFTWARE DATA PROTECTION 29
APPENDIX C: DIAGNOSTIC MODE 30

JTEC Model X-Port10
Intelligent CAMAC-to-FERA Bridge

1. OVERVIEW

JTEC Model X-Port10 is an intelligent, programmable, and re-configurable auxiliary CAMAC controller designed for reading out standard CAMAC modules onto a FERA bus, an ECL-port protocol designed by LeCroy. The module is intelligent in that it executes sequences of operations on both CAMAC and ECL buses and makes logical decisions without intervention by an external CPU. It is programmable in that the user programs it, via issuing to it a number of CAMAC commands, to perform the readout of a particular set of CAMAC modules in a manner suited for a particular experiment. And lastly, X-Port10 is re-configurable in that one can change its functionality by loading different configuration data into its Field Programmable Gate Array (FPGA). Up to 4 configuration files can be stored in the on-board in-site-re-programmable flash memory.

More accurately, X-Port10 is a universal logical module with 5 ECL input ports, 6 direct ECL output ports, and 16 ECL output ports mediated by a 16-bit FIFO (First-In-First-Out memory).

1.1. FEATURES

- Auxiliary crate controller in a single-width CAMAC module, with full auxiliary port arbitration.
- Built-in CAMAC module definition database stores the A and F codes for Q-test, readout, and clearing functions for different module types.
- Up to 8 CAMAC module types can be defined by the user and used along with the built-in database.
- Programmable CAMAC crate setup information includes among other things: slot numbers occupied by various modules; the types of modules residing in these slots; the number of channels to be read out from the individual modules; a delay time from the DAQ system trigger to the start of the readout; an optional LAM timeout period.
- Complete support for the readout of XIA digital pulse processors (DPP), the DXP-4C and DGF-4C.
- Capability to handle a hit-register-based sparse readout, as implemented in the Phillips CAMAC modules.
- Support for standard CAMAC (3MB/sec), Level-1 (7.5Mb/sec) and, optionally, Level-2 (60MB/sec) FASTCAMAC.
- Optional Q-test for individual modules to skip the empty modules.

- Optional wait-for-LAM to start the readout of individual modules.
- Optional triggering by a selection of LAMs (LAM mask).
- Fast clear of the X-Port10 is possible after the DAQ trigger is received but the event is found unworthy acquiring.
- Intermediate storage of data in fast, dual-port FIFOs for the subsequent asynchronous delivery via the ECL port.
- Concurrent readout of multiple CAMAC crates by their individual X-Port10s.
- FIFOs are socketed for easy depth customization, up to a maximum of 192K of 16-bit words.
- FERA ECL port frequency up to 10 MHz, dependent on the remote FERA controller.
- Busy In and Busy Out ports for a convenient generation, via daisy-chaining, of a logical OR of busy signals of all X-Port10 modules used in the setup.
- The firmware is implemented in a field-programmable gate array (FPGA), and can be customized to satisfy special requirements of particular experiments.
- The configuration flash memory holds four complete configurations that can be selected with two jumpers.
- The configuration flash memory can be reprogrammed via CAMAC.

2. SPECIFICATIONS

2.1. CAMAC PROTOCOL

2.1.1. CONTROL BUS

Connector: 88 – finger card-edge

Address I/O (A1 – A4): Camac Address.

Function I/O (F1 – F5): Camac Function.

1st Strobe I/O (S1): Camac strobe. Indicates the presence of valid data on the read/write data bus.

2nd Strobe I/O (S2): Camac strobe

Valid Command I/O (X): Camac X. Indicates that the command received is a valid one/

Valid Data I/O (Q): Camac Q.

Crate Inhibit

Input (I): Camac Inhibit.

Slot Address
Input (N): Camac N.

2.1.2. CAMAC DATA BUS USAGE

Read I/O (R): 24 CAMAC “read” lines.
Write I/O (W): 16 CAMAC “write” lines.

NOTE: some of the ports characterized as I/O in the above specifications are not necessarily implemented as such in any particular FPGA configuration. For example, the Camac “write” port W1 – W16 may be used as an input and not as an input/output port.

2.1.3. CAMAC AUXILIARY BUS

Connector: 2 x 20 pin rear connector.

Input/output levels: Open Collector TTL; negative logic.

Slot Address
Output (N_AUX): 5 lines.

Aux. Control Lockout
Output (ACL): 1 line.

Aux. Request Inhibit
I/O (ARI): 1 line.

Aux. Request Output
(ARQ): 1 line.

Look At Me Input (L): 24 lines from individual CAMAC slots.

2.1.4. AUXILIARY BUS ARBITRATION

Connectors: 3 front-panel LEMO

Input/output Levels: Open Collector TTL; negative logic.

Aux. Request Output
(RQ): 1 connector; parallel with the rear connector ARQ line. Generated when access to CAMAC modules is required.

Grant In Input
(GI): 1 connector. 470 Ω pull-up to 5V. Indicates that the module is allowed to take control of the CAMAC crate.

Grant Out Output (GO): 1 connector. Echoes the GI input when the (intrinsic) ARQ is not present.

2.1.5. CAMAC COMMANDS

CAMAC commands are described in detail in Section 4. They are also summarized in Appendix A.

2.2. FERA PROTOCOL

2.2.1. ECL COMMAND BUS

Connector: 8 x 2-pin front panel connector. The input-matching and output pull-down resistors may be removed for high-impedance operation. The presence of these resistors is indicated by a front-panel yellow LED.

Input Levels: Differential ECL, 100 Ω differential impedance.

Output Levels: Differential ECL into 100 Ω differential.

Clear Input (CLR): Minimum width 20 ns.

Request Output (REQ): ECL port readout request. Is set only when the module is ready to deliver data to the output port. Shared with diagnostic “strobe” monitor output.

Write Strobe Output (WST): Indicates the presence of valid data on the ECL output port. The ECL port data is stable during entire WST pulse. Is released when WAK is received. Shared with diagnostic FIFO “write” monitor output.

Write Acknowledge Input (WAK) Acknowledge signal from the ECL port receiver indicating that data has been loaded.

2.2.2. ECL PORT ENABLE/PASS

Gate Input (GA): 1 x 2 – pin front-panel connector. Impedance 100 Ω differential. Minimum width 20 ns.

Readout Enable Input (REN): 1 x 2 – pin front-panel connector. Impedance 100 Ω differential.Must be maintained by the ECL port receiver for the duration of the readout.

Pass Output (PASS): 1 x 2 – pin front-panel connector. Differential ECL into 100 Ω differential. Echoes REN when (intrinsic) REQ is not present. Shared with diagnostic FIFO data bit monitor output.

2.2.3. ECL PORT OUTPUT

Connector: 17 x 2-pin front-panel connector; the last 2 pins are not connected.

Output levels: Differential ECL into 100 Ω differential.
The pull down resistors may be removed for high impedance outputs. The presence of these resistors is indicated by a yellow front panel LED.

Specifications: Data word size: 16 bits. Sequential data readout with maximum output frequency 10 MHz.

2.3. VARIOUS ECL CONTROL PORT SIGNALS

Outputs UO1-UO4: four 1 x 2 – pin front-panel connectors. Impedance 100 Ω differential. Functionality defined by the firmware.

Input UI1: 1 x 2 – pin front panel connector. Impedance 100 Ω differential. Functionality defined by the firmware.

2.4. INTERNAL DATA STORAGE FIFO

FIFO width: 16 bits (two FIFO chips).

FIFO depth: Up to three pairs of FIFO chips. Minimum - 1 pair.

FIFO model: CY9C421-20, CY9C433-20 by CYPRESS, or equivalent.

2.5. POWER REQUIREMENTS

+6V:	Approximately 430 mA
-6V:	Approximately 360 mA

3. FUNCTIONAL DESCRIPTION

The functionality of X-Port10 is largely determined by the firmware loaded into its field-programmable gate array, an XCS30 of the Spartan series by XILINX. At the same time, it is limited by the capacity of the XCS30 FPGA and by the physical layout of the module. The most fundamental function of X-Port10 is the readout (pursuant to a list programmed into it by the user) of a number of CAMAC modules into its internal dual-port first-in-first-out memories (FIFO) and an asynchronous transfer of the FIFO content, via the ECL data output port, to a remote ECL port receiver. The readout of CAMAC modules proceeds according to the CAMAC protocol, while the ECL transfer proceeds according to the FERA protocol developed by LeCroy Research Systems Corp. A readout/transfer cycle is triggered either by a gate signal provided by the DAQ system or, optionally, by a user-defined pattern of LAM signals from the serviced CAMAC modules. Upon completion of a readout/transfer cycle, X-Port10 returns to the idle state, ready to process the next event.

The use of FIFOs enables a concurrent readout of multiple CAMAC crates by their individual X-Port10 controllers, which results in an overall event processing time that is significantly shorter than allowed by the standard sequential CAMAC readout protocol.

3.1. FIRMWARE VERSIONS

The controlling FPGA configures itself upon power-up from an on-board flash memory, a Programmable Erasable Read Only Memory (PEROM) AT29C010A by ATMEL. This flash memory accommodates four full XCS30 configurations, each occupying first 30996 bytes of a 32 KB segment of the 128 KB memory. The active section and, hence, the configuration to be loaded is selected by two jumpers JP12 on the X-Port10 board. At the time of the release of the present document, there are two versions of X-Port10 firmware, the Standard Version STD24 and the XIA Digital Pulse Processor Version DPP.

3.1.1. THE STANDARD VERSION

The Standard Version of the X-Port10 firmware is designed to handle various types of commercial CAMAC modules, such as ADCs, QDCs, TDCs, Scalers, Bit Registers, etc. by LeCroy, ORTEC, Phillips Scientific, and others. This version is identified by the Version ID = 2225 hexadecimal. The version ID can be read upon power-up via the CAMAC command **NF(0)A(12)**.

Note, that the relevant jumper setting and the revision number may change, as the flash memory is reprogrammed. Hence, it is advisable to always identify the actual version and revision numbers by reading the Version ID using **NF(0)A(12)** (see 4.3.2).

3.1.2. THE DPP VERSION

The DPP version of the X-Port10 firmware is dedicated to handling digital pulse processors by XIA, including the Digital Gamma Finder (DGF-4C) and Digital X-ray Processor (DXP-4C). This version is identified by the Version ID = 8F02 hexadecimal.
Note: Always identify the actual firmware by reading its ID, using the CAMAC command **NF(0)A(12)** (see 4.3.2).

3.2. CAMAC MODULE TYPES AND INDIVIDUAL MODES OF OPERATION

X-Port10 distinguishes CAMAC modules according to the CAMAC **F(X)A(Y)**codes that are needed to handle their readout and subsequent clearing. These **F(X)A(Y)** codes are stored in a read-only database of X-Port10 and in the RAM area for the user-defined CAMAC modules. In its first access of a CAMAC module, X-Port10 queries, depending on the type of the module accessed, either (i) about addresses of module registers (hit-pattern) that contain valid data, (ii) the number of Cssed tobed subsequennlyfetchnedfro(m)8.4 -

[1] The non-zero bits of the HitPattern word point to module registers that contain valid data. For example, a HitPattern = 0009h (hexadecimal) results in X-Port10 generating two "Read" commands, **NF(0)A(0)** and **NF(0)A(3)** (note that the first bit corresponds to address 0).

[2] This number includes the result of the query itself. For example, NumberOfWords = 10 indicates that there are 9 data words to be read out using the CAMAC command **NF(0)A(0)**.

[3] The readout proceeds according to the FastCamac protocol. The internal pointer of a DPP module is incremented upon CAMAC **S1** and the sequence is terminated by a CAMAC **S2**. **NF(5)A(0)** is sustained continuously for the duration of the sequence.

Type 0 includes ADC and TDC modules by Phillips Scientific – models 7164(H), 7166(H), 7167(H), and 7186(H).

Type 1 includes ADCs and TDCs by LeCroy - models 2249A, 2249SG, 2249W, 2259B, 2228A, and 2229. It includes also model 2551 scaler by LeCroy.

Type 2 includes the AD811 ADC by Ortec.

Type 3 includes the 4448 coincidence register by LeCroy.

Type 4 includes the 2341S and 2341A coincidence registers by LeCroy.

Type 5 includes the 4434 latching scaler by LeCroy.

3.2.1. USER-DEFINED MODULES

Revision 0 of the Standard Version supports user-defined CAMAC modules with Q-test (i.e., functionally similar to types 1 and 2, except of using different CAMAC function codes). The codes for the initial query, register read, and clearing are programmable by the user using CAMAC function **NF(16)A(11)W(FACodes)** (see 4.2.8).

3.2.2. INDIVIDUAL MODULE MODES OF OPERATION

There are two modes of conditional module readout that can be set on the individual basis for each module – the Q-test mode and the LAM-test ("wait-for-LAM") mode (see 4.2.3).

3.2.2.1. 24-BIT READOUT MODE

When programmed for 24-bit readout, X-Port10 fetches a 24-bit word from the CAMAC module and stores it in its FIFOs as one 16-bit and one 8-bit word, for a subsequent transfer to the FERA receiver.

3.2.2.2. LAM-TEST MODE

When programmed for operation in LAM-test mode, the beginning of the module readout is delayed until the module has set LAM. Should the module fail to set LAM within a (global) programmed LAM timeout period (see 4.3), its readout is suppressed.

3.2.2.3. CLEAR-SUPPRESS MODE

When programmed for operation in Clear-suppress mode, X-Port10 suppresses clearing of the CAMAC module.

3.3. GLOBAL MODES OF OPERATION

X-Port10 can be operated in several different modes by selecting among two available trigger options, three options for the clearing of CAMAC modules, or by selecting a diagnostic option. The selection of different modes is achieved by setting proper bits in the global mode parameter (see 4.2.5).

3.3.1. TRIGGER MODES AND TIMER SETTINGS

By default, X-Port10 is triggered by an external trigger signal at its ECL control port GA. Optionally, X-Port10 may be triggered by a valid LAM pattern as prescribed by the content of the LAM mask (see 4.2.5. and 4.2.7).

When triggered by the GA signal, X-Port10 waits for a preprogrammed time interval to allow conversion process in the CAMAC modules to complete and then begins the readout process. The amount of trigger delay time is a global parameter, programmable in 1μs increments (see 4.2.6).

The response to the valid LAM pattern, assuming that the LAM-based trigger option is selected, depends on the version of the firmware. In the DPP Version, X-Port10 is triggered by any of the active LAM signals as defined by the non-zero bits of the LAM mask (see 4.2.7). However, the readout cycle begins only when all active LAM signals are set within the LAM timeout period. When one or more LAMs are missing by the lapse of the LAM timeout, X-Port10 clears the LAMs of all modules and aborts the readout. The LAM timeout period is programmable in 1μs increments (see 4.2.6).

In the Standard Version, X-Port10 begins the readout/transfer cycle right after master LAM (defined by 5 least significant bits of the LAM mask) is detected. In this case the LAM timeout determines the maximum time interval X-Port10 will wait for any LAM-

tested module (see 3.2.2.2) to set its LAM. Should the module fail to set LAM, its readout is suppressed.

3.3.2. CAMAC MODULE CLEARING OPTIONS

Upon completion of an event handling, X-Port10, by default, clears all CAMAC modules of type 1 - 4 by individual addressed CAMAC commands, as shown in Table 1. It generates also at its ECL control port UO1 a signal suitable for clearing all modules of type 0. Optionally, modules of type 1 - 4 can be cleared collectively by a CAMAC crate clear command **C•S2**, or clearing can be suppressed altogether. In the latter case, it is up to the data acquisition system to provide for means of clearing for all modules. Modules of the DPP type are expected to perform proper clearing of their data memory area on their own, upon detecting an end of the readout/transfer process (e.g., by monitoring the "busy" signal at the ECL UO4 control port, or by monitoring the signal at the ECL UO1 control port).

NOTE: There is no Clear option for modules of type 5 (latching scaler by LeCroy).

WARNING: while considering collective clearing of modules by **C•S2**, one should keep in mind that for some modules (e.g., of type 0) the **C•S2** command clears module configuration registers (along with the data registers), necessitating reloading of these registers after every event – an option that seems to make little, if any sense.

3.3.3. Virtual Station Number

Virtual Station Number identifies a particular X-Port10

The first word of the data block for an individual CAMAC module contains information on the number of data words in the block. For the Standard Version of the X-Port10 firmware, this word has a form of a 16-bit hit pattern, where bits set to 1 identify addresses of module registers from which the subsequent data words in the block were fetched from. For the DPP Version of the firmware, the first word in the data block for an individual DPP module represents the total number of words in the block.

4. OPERATING INSTRUCTIONS

Successful operation of X-Port10 requires a proper hardware setup and the programming of the operating parameters via CAMAC interface.

4.1. HARDWARE SETUP

The hardware setup includes (i) configuring ECL port pull-down and impedance-matching resistor arrays, (ii) selecting the desired version of the operating firmware, (iii)

setting up the operation of storage FIFOs, (iv) connecting to the CAMAC bus, (v) connecting the auxiliary port of X-Port10 to the auxiliary port of the main CAMAC controller, (vi) connecting the auxiliary controller arbitration ports to the arbitration daisy-chain, and (vii) connecting ECL ports to ECL data and control lines,

4.1.1. ECL PORT PULL-DOWN AND IMPEDANCE MATCHING RESISTORS

The ECL port pull-down and impedance-matching resistor arrays should be in place only when X-Port10 is the last device on the FERA bus. Otherwise, they should be removed and stored in the blind sockets of the dedicated "parking" area on the X-Port10 board.

Note that there are two positions available for the resistor arrays associated with the ECL control port input lines. The left socket is for WAK and CLR lines only, while the right socket includes additionally the GA (external trigger) line.

4.1.2. FIRMWARE VERSION SELECTION

The operating firmware is selected by setting two jumpers 1 and 2 of JP12 on the X-Port10 board. These jumpers can be accessed via a cut-out in the side panel of X-Port10. The right-most position of the jumper is idle, while the leftmost sets the bit of the 2-bit sector ID – top jumper sets bit 1, while the bottom jumper sets bit 2:
Sector 0 – both jumpers in right-most positions.
Sector 1 – top jumper in left-most position, bottom one in right-most positions.
Sector 2 – top jumper in right-most position, bottom one in left-most positions.
Sector 3 – both jumpers in left-most positions.

4.1.3. FIFO SETUP

There are three pairs of FIFO sockets (PLCC) on the X-Port10 board, labeled **A**, **B**, and **C**. The operation of X-Port10 requires that at least one pair, the pair **A**, be populated by FIFO ICs of desired/available capacity (see the technical notes below). While the populating of the remaining sockets is optional, they must be populated in the "natural" order – first **B** and then **C**. The FIFO socket population must be matched by the proper setting of the "FIFO select" jumper JP13. The presence of the pair **A** must be matched by the setting of the single jumper in the position **AA**, the presence of pairs **A** and **B** must be matched by the setting of the jumper in the position **AABB**, and, lastly, the presence of pairs **A**, **B**, and **C** must be matched by the setting of the jumper in the position **AABBCC**.

TECHNICAL NOTE 1: X-PORT-10 first stores the data read from the CAMAC modules in its FIFOs. Subsequently and asynchronously with respect to the "FIFO Write" operations, the content of the FIFO is presented to the ECL data output port. Hence, the presence of FIFO(s) is absolutely essential for the operation of X-PORT-10.

TECHNICAL NOTE 2: The jumper JP13 serves to route the depth expansion signals of the FIFO ICs. Accordingly, an improper setting of this jumper results in a FIFO malfunction.

While the capacity of the on-board FIFO(s) can be chosen arbitrarily, an optimum choice is the one that guarantees the fastest possible event processing. As the FIFO is completely read out (and, hence, cleared) after every individual event processing, there is no advantage of having the FIFO capacity larger than needed for the storage of one full event. Also, in the case where X-Port10 is the first device to gain the access to the FERA bus, there is no advantage of having a FIFO capacity larger than the minimum capacity available. This is so, because the readout of the FIFO proceeds concurrently with the writing into it of the CAMAC data and because the latter operation proceeds at a lower pace than the FIFO readout process. As a result, in such a case, at no moment in time will there be more than one word stored in the FIFO.

4.1.4. CONNECTING TO CAMAC BUS

X-Port10 is connected to the CAMAC bus by its insertion into a selected slot in the CAMAC crate.

4.1.5. AUXILIARY PORT CONNECTOR

The rear 40-pin auxiliary CAMAC port of X-Port10 is to be connected by a ribbon cable to the auxiliary port of the main CAMAC controller. Pin 1 of this rear connector is located at the bottom.

4.1.6. AUXILIARY CONTROLLER ARBITRATION

Auxiliary controller arbitration utilizes two lines of the auxiliary CAMAC port as well as from one to three front-panel LEMO ports. For the operation of X-Port10 it is essential that it is granted control of the auxiliary bus when it is ready to perform CAMAC operations and has requested permission for such control. The request signal is asserted on ARQ line of the auxiliary CAMAC bus and, in parallel, at the front-panel ARQ LEMO port (TTL, negative logic). The module accepts the grant of control via the front-panel AGI port (TTL, negative logic). It releases the control by retransmitting the Auxiliary Grant In signal over the Auxiliary Grant Out (AGO) front-panel LEMO port to a subsequent device of a lower priority.

For a proper bus arbitration, the Grant In and Grant Out ports (in the case of X-Port10, the AGI and AGO ports) of the controllers must be daisy-chained on a priority-based manner. The Grant In port of the highest priority module is to be fed directly from the Request port (in the case of X-Port10, the ARQ port), while the Grant In port of any other lower priority controller is fed from the Grant Out port of a controller that has the priority higher by one level.

4.1.7. BOOTING AND REBOOTING

X-Port10 boots automatically, upon power-up, from the active segment (see 4.1.2) of the flash memory (an AT29C010A by ATMEL). It is possible to reboot X-Port10, without cycling the power, by issuing the sequence of two CAMAC commands: **NF(19)A(5)W(1)** followed by **NF(19)A(5)W(2)**.

5. PROGRAMMING OF X-Port10

The operation of X-Port10 requires its prior programming with a set of operating parameters describing the configuration of the CAMAC crate and the desired mode of operation. The programming consists of a series of CAMAC "Write" commands **NF(16)A(X)W(Data)**, where **X** identifies the parameter and **Data** is a 16-bit word containing the parameter value presented in a proper format. The programming sequence is arbitrary, with one important exception that the individual CAMAC module description commands, **NF(16)A(2)W(ModuleData),** must follow the command specifying the number of modules to be serviced, **NF(16)A(1)W(NumberOfModules)** (see the technical note below)**.**

TECHNICAL NOTE: The command **NF(16)A(1)** resets the pointer of the configuration memory to point to the first virtual CAMAC module data. This pointer is then incremented with every **NF(16)A(2)** command, allowing one to program data for consecutive modules.

X-Port10 responds to every valid CAMAC command by X=1.

5.1.1. NF(16)A(0)W(VSN) – VIRTUAL STATION NUMBER

Any FERA compatible device is expected to identify itself to the ECL output port receiver by a 16-bit Virtual Station Number VSN. This number is programmed into X-Port10 by a CAMAC command NF(16)A(0)W(VSN), where W(VSN) has the following simple format:

16	**1**
Virtual Station Number	

The VSN value is not essential for the operation of X-Port10. It is, however, required by the FERA protocol that every device on the FERA bus be identified by its unique VSN.

EXAMPLE: **NF(16)A(0)W(8317h)** sets VSN=8317 hexadecimal.

5.1.2. NF(16)A(1)W(NumberOfModules) - NUMBER OF CAMAC MODULES

One X-Port10 controller can service up to 31 "virtual" CAMAC modules, i.e., more than can be physically accommodated in one CAMAC crate. This is so, because one CAMAC module may require several different CAMAC functions for the complete readout of its registers. For, example, part of the registers may be read via the function **NF(0)A(0-15)**, while other part may be read via the function **NF(1)A(0-15).** In that case, X-Port10

considers one physical device as two virtual modules occupying the same slot in the CAMAC crate. The relevant "Write" data word has the following structure:

16 6	5 1
Unused	**Number of Modules**

where Number Of Modules = 1 – 31 decimal.

EXAMPLE: **NF(16)A(1)W(11h)** instructs X-Port10 to process 17 "virtual" CAMAC modules.

5.1.3. NF(16)A(2)W(ModuleData) - CAMAC MODULE DESCRIPTION

The ModuleData is a composite 16-bit "write" data word containing the information on the CAMAC module slot number, Q-test and LAM-test options, module type, and, where applicable, the last register address to be read. The ModuleData word has the following structure:

16 13	12 9	8	7	6	5 1
Last Address[6)]	**Module Type[5)]**	**No-Clr[4)]**	**LAM-test[3)]**	**24-bit [2)]**	**Slot Number[1)]**

[1)] Slot Number = 1 – 24. Mandatory.

[2)] 24-bit = 1 causes X-Port10 to fetch 24-bit words from the CAMAC module and to store them as one 16-bit and one 8-bit words, for subsequent transfer to the FERA receiver Not applicable to the DPP version.

[3)] LAM-test = 1 (0), turns on (off) the module readout suppression whenever the module has failed to set LAM within the LAM timeout period. Not applicable to the DPP Version.

[4)] No-Clr=1 suppresses clearing of modules of type 0-4. For type 5, No-Clr=1 is implied and bit 8 represents 5th bit of the 5-bit Last Address word (bits 13-16 representing the 4 least significant bits).

[5)] Module Type is as specified in Table 1. Not applicable to the DPP Version.

[6)] LastAddress = 0 – 15. X-Port10 reads addresses 0 through LastAddress. Not applicable to modules of type 0 and to the DPP Version. Note that modules of type 5 have 5-bit LastAddress word, the 5th bit being stored in bit 8 of the Module Description word.

EXAMPLE: **NF(16)A(2)W(5223h)** instructs X-Port10 to read addresses 0 through 5 for an ORTEC AD811 ADC residing in the third slot of the CAMAC crate, on the condition that the module has responded with Q = 1 to the query command **NF(8)A(12).**

5.1.4. NF(16)A(3 – 6)W(DelaySubtractor) – "FIFO WRITE" DELAY

Any CAMAC module is expected to respond to a "read" command by supplying stable data on the CAMAC R, Q, and X lines by the time of the first CAMAC strobe S1, i.e., 400ns after it has received a valid **NF(X)A(Y)** command. Accordingly, X-Port10 generates by default a 400ns delay, before it writes the state of the CAMAC R-lines (the data) into its internal FIFOs. However, most of the modern CAMAC modules provide for stable data on the CAMAC bus significantly earlier than 400ns. To speed up the CAMAC readout process, the user may find it desirable to fine-tune the delay time by module types, by amounts programmed as DelaySubtractors. These amounts are multiplied by 40ns and subtracted from the default value of 400ns. For example, a delay subtractor value of 5 results in X-Port10 generating an internal "FIFO Write" signal 200ns, and not 400ns, after it has issued the CAMAC "Read" command. The subtractors are three bit numbers and are stored in the 16-bit data word in a format shown in the diagram below.

16	15 – 13	12	11 – 9	8	7 – 5	4	3 – 1
	DelSubtrD		**DelSubtrC**		**DelSubtrB**		**DelSubtrA**

In the case of the Standard Version, The CAMAC command NF(16)A(3) stores the delay subtractors for CAMAC modules of types 0 through 3, NF(16)A(4) – for types 4 through 7 (these types are not implemented in Revision 0), NF(16)A(5) – for user-defined types 8 through 11, and NF(16)A(6) – for user-defined types 12 through 15.

In the case of the DPP Version, only **NF(16)A(3)** is applicable, with **DelSubtrA** being the only subtractor.

EXAMPLE: **NF(16)A(3)W(0331h)** reduces the delays for the "FIFO Write" signals by 40ns, 120ns, and 120ns, for the modules of type 0 (Phillips type), type 1 (LeCroy 22xx type), and type 2 (ORTEC AD811), respectively.

5.1.5. NF(16)A(7)W(GlobalMode) - GLOBAL MODE OF OPERATION

The GlobaMode parameter is a composite 16-bit data word encoding the information on the CAMAC module clearing option, diagnostic mode option, diagnostic data bit multiplexer, and X-Port10 trigger mode. The word has the following structure:

16 – 14	13	12 – 9	8 – 6	5	4 – 3	2 – 1
Unused	**Trigger Mode[4)]**	**Diagnostic Multiplexer[3)]**	**Unused**	**Diagnostic Mode[2)]**	**Unused**	**Clearing Mode[1)]**

[1)]Clearing Mode = 0 (default) – individual clearing of modules via addressed CAMAC commands **NF(9)A(0)** (type 1), or **NF(11)A(12)** (type 2).
Clearing Mode = 1 – no module clearing (the data acquisition system is assumed to perform module clearing at the end of every event).

Clearing Mode = 2 – all modules are cleared simultaneously via **C•S2**.
Not applicable to the DPP Version.

[2)]Diagnostic Mode = 1 (0) activates (deactivates) diagnostic mode, an advanced feature described in Appendix C.

[3)]An advanced feature described in Appendix C.

[4)]Trigger Mode = 0 (default) causes X-Port10 to commence event processing upon receipt of the GA signal via the front-panel ECL control port.
Trigger Mode = 1 causes X-Port10 to commence event processing upon the receipt of the valid LAM signals (see 4.2.7.).

EXAMPLE 1: **NF(16)A(7)W(0)** instructs X-Port10 to respond to the GA trigger signal.

EXAMPLE 2: **NF(16)A(1000h)** instructs X-Port10 to respond to a valid combination of LAM signals.

5.1.6. NF(16)A(8)W(Delays) – TRIGGER AND LAM TIMING

The word Delays is a composite 16 bit word containing information on the LAM timeout period and on the desired delay measured from the GA trigger signal to the start of the CAMAC modules readout. The purpose of the latter delay is to allow the CAMAC modules to complete conversion, before the readout is initiated. LAM timeout plays somewhat different roles in the two versions of the X-Port10 firmware. In the Standard Version, it is used in conjunction with the LAM-test option for an individual module (see 4.2.3). In this case, the readout of an individual LAM-tested module is suppressed whenever the module fails to set LAM within the timeout period (counted from the trigger signal). In the DPP Version, a LAM fault is declared whenever any of the requested LAMs (as defined by the LAM mask – see 4.2.7) is not set within the LAM timeout period. In that case, the event readout is suppressed altogether. However, X-Port10 still generates the individual **NF(1)A(2)** commands for all DPP modules, to clear all LAMs that were set.

The word Delays has the following structure:

16	**9**	**8**	**1**
TriggerDelay		**LAMTimeout**	

The time is measured in units of 1μs.

EXAMPLE: **NF(16)A(8)W(8020h)** sets trigger delay to 128μs and LAM timeout to 32μs.

5.1.7. NF(16)A(9, 10)W(LAMMaskX) – LAM MASK

The LAM mask word is used in conjunction with the global trigger option TriggerOption = 1 (see 4.2.5) and it identifies the CAMAC slots housing LAM-enabled modules.

In the Standard Version of the X-Port10 firmware, the LAM mask word is 5 bits long and represents the slot number of the master module, i.e., the module whose LAM triggers the event readout. In this version, the command **NF(16)A(10)** is inapplicable and the LAMMask word has the following structure:

16	**6**	**5**	**1**
Unused		**MasterLAM**	

In the DPP Version, the LAM mask is 24 bits long and, hence, its programming requires two different CAMAC functions. The function **NF(16)A(9)W(LAMMask1)** programs the first twelve bits and the function **NF(16)A(10)W(LAMMask2)** programs the last twelve bits of the LAM mask according to the following format:

16	**13**	**12**	**1**
Unused		**LAMMaskX**	

EXAMPLE 1: For the Standard Version and TriggerOption = 1, the **NF(16)A(9)W(5)** command sets X-Port10 to trigger upon LAM set by the module residing in the fifth slot.

EXAMPLE2: For the DPP Version and TriggerOption = 1, the two commands **NF(16)A(9)W(*111h)** and **NF(16)A(10)W(*111h)** ("*" indicates "don't care") sets X-Port10 to commence a readout/transfer cycle whenever LAMs are set for slots 1, 5, 9, 13, 17, and 21. The readout is suppressed when some, but not all of the active LAMs are set within the LAM timeout period.

5.1.8. NF(16)A(11)W(FACodes) – DEFINE A NEW MODULE TYPE

16	**15**	**14**	**13**	**12**	**9**	**8**	**5**	**4**	**1**
Unused		**CommandType[4)]**		**F4 – F1[3)]**		**A4 – A1[2)]**		**ModuleType[1)]**	

[1)]User-defined ModuleType = 8 – 15.

[2)]Not applicable to CommandType = 2 (see comment 4, below).

[3)]Valid function codes are F = 0 – 15.

[4)]CommandType = 0 – Q-test (module query) command.
CommandType = 1 – module clearing command.
CommandType = 2 – data register reading command.

EXAMPLE: a sequence of three commands: **NF(16)A(11)W(628h), NF(16)W(1908h), NF(16)W(2108h)** defines type 8 of a CAMAC module, for which the Q-test command is $\mathbf{N_{module}F(6)A(2)}$, the clearing command is $\mathbf{N_{module}F(9)A(0)}$, and the data register "Read" command is $\mathbf{N_{module}F(1)A(0 - 15)}$.

5.2. NF(0)A(X) – READ-BACK OF THE SETUP DATA

X-Port10 allows one to read back the operating parameters it is programmed with. It allows one also to read the content of a 16-bit read-only register containing the firmware ID. The data read back have the same format as was used for their programming. All fields of the type "don't care" (or "inapplicable") appear in the read-back data as zeroes.

5.2.1. READ-BACK OF OPERATING PARAMETERS

The CAMAC function code for the parameter read-back is 0 and the address code is the same as used in the prior programming of the parameter of interest. For example, the command **NF(0)A(1)** returns the number of virtual CAMAC modules serviced, previously programmed via the command **NF(16)A(1)**. The read-back is not possible for the CAMAC codes of the user-defined modules, programmed via the commands **NF(16)A(11)**. Otherwise, it can be performed in an arbitrary sequence with one important exception that the read-back of the module data via consecutive **NF(0)A(2)** commands must be preceded by the read-back of the number of virtual modules via **NF(0)A(1)** (see the technical note below).

TECHNICAL NOTE: The command **NF(0)A(1)** resets the pointer of the configuration memory to point to the first virtual CAMAC module data. This pointer is then incremented with every **NF(0)A(2)** command, allowing one to retrieve the data for consecutive modules.

5.2.2. NF(0)A(12) - FIRMWARE ID

The CAMAC function **NF(0)A(12)** returns the ID code of the operating firmware. The ID word has the following format:

16 13	**12 9**	**8 5**	**4 1**
CreationMonth	**CreationYear[2)]**	**RevisionNumber**	**VersionNumber[1)]**

[1)]VersionNumber = 1 – Standard Version
VersionNumber = 2 – DPP Version

[2)]CreationYear = Fh (decimal 15) – 1999

EXAMPLE 1: Firmware ID = 2225h indicates rev. 2 of the 24-bit Standard Version (STD24) created in February 2002.

EXAMPLE 2: Firmware ID = 8F02h indicates rev. 0 of the DPP Version created in August 1999.

5.3. EXAMPLES OF X-Port10 PROGRAMMING

5.3.1. EXAMPLE 1 – STANDARD VERSION

The following sequence of CAMAC commands sets up X-Port10 for reading a Phillips ADC residing in slot 1, a LeCroy Model 2249 QDC residing in slot 3, and an ORTEC AD811 residing in slot 5:

NF(16)A(0)W(8555h)	VSN=8555 hexadecimal
NF(16)A(8)W(8080)	start readout 128μs after being triggered; set LAM timeout to 128μs as well
NF(16)A(1)W(3)	3 modules to be serviced
NF(16)A(2)W(1)	module of type 0 in slot 1
NF(16)A(2)W(2123h)	read channels 0 through 2 from a module of type 1 residing in slot 3. Use 24-bit readout.
NF(16)A(2)W(3245h)	read channels 0 through 3 from an ORTEC AD811 residing in slot 5. Wait for LAM and suppress readout if no LAM is set within the LAM timeout period of 128μs

Note that, by default, X-Port10 will trigger on external GA signal. Also by default, the three modules will be cleared upon the completion of readout by their respective "Clear" commands.

5.3.2. EXAMPLE 2 – STANDARD VERSION

The following sequence of CAMAC commands sets up X-Port10 for reading out a two identical user defined CAMAC modules for which the "Test Q" command is $\mathbf{N_{module}F(6)A(2)}$, the "Clear" command is $\mathbf{N_{module}F(9)A(2)}$, and the "Read" command is $\mathbf{N_{module}F(2)A(0\text{-}15)}$:

NF(16)A(0)W(8555h)	VSN=8555 hexadecimal
NF(16)A(11)W(628h)	define the "Test Q" command for a module of type 8 to be $\mathbf{N_{module}F(6)A(2)}$
NF(16)A(11)W(1928h)	define the "Clear" command for a module of type 8 to be $\mathbf{N_{module}F(9)A(2)}$
NF(16)A(11)W(2208h)	define the "Read" command for a module of type 8 to be $\mathbf{N_{module}F(2)A(0\text{-}15)}$
NF(16)A(7)W(1000h)	trigger on master LAM
NF(16)A(9)W(2)	master LAM from slot 2
NF(16)A(8)W(80)	set LAM timeout to 128μs
NF(16)A(1)W(2)	two modules to be serviced
NF(16)A(2)W(F802h)	read channels 0 through 15 for a module of type 8 residing in slot 2.

NF(16)A(2)W(B843h)	read channels 0 through 11 for a module of type 8 residing in slot 3. Suppress readout if the module has not set LAM.

Note that the trigger delay timeout is not used in this particular setup. By default, the modules will be cleared upon the completion of readout by their individual $\mathbf{N_{module}F(9)A(2)}$ commands.

6. STRUCTURE OF THE OUTPUT DATA STREAM

In every valid event, X-Port10 outputs to the ECL data stream first its Virtual Station Number VSN and then blocks of data corresponding to individual CAMAC modules. Accordingly, the segment of the data buffer written has the following general structure:

VSN
Data block for the first module
Data block for the second module
.....
.....
Data block for the last module

The data block for each CAMAC module begins either with a hit pattern (Standard Version – modules of type 0), a number specifying the number of data words output onto the FERA data bus (Standard Version – modules of type 1 – 5), or a number specifying the number of data words in the block (DPP Version). In all cases, the first data word in the block defines the length of the block in an unambiguous manner. Note that the non-zero bits of the hit pattern identify the register addresses (or module channels) from which the subsequent data were fetched. Hence, for the modules of type 0 in Standard Version, the module data block has the structure:

Hit Pattern
Data corresponding to the module channel identified by the lowest bit set
Data corresponding to the module channel identified by the second lowest bit set
.....
.....
Data corresponding to the module channel identified by the highest bit set

For modules of type 1- 5 in Standard Version, the data block has the structure:

Number of data words to follow
1st data word
2nd data word
.....
.....
last data word

The structure of the data block for a DPP module is as follows:

Number of words in the block (including the first word)
1st data word
2nd data word
.....
.....

last data word

As is clear from the above description, the composition of the buffer segment generated by X-Port10 depends on the firmware version and the actual setup, which must be accounted for by the buffer unpacking routine.

EXAMPLE: In a setup as considered in the Example 1 in 5.3.1 and assuming that the first module had a hit pattern 0005h, the buffer segment written by X-Port10 looks as follows:

8555h	VSN
5	hit pattern for 1st module of type 0 (bits 1 and 3 set)
data from the 1st channel	
data from the 3rd channel	
6	6 ECL words for three module data words (24-bit readout)

1st data word, bits 1-16 of module data
2nd data word, bits 17-24 of module data
3rd data word, bits 1-16 of module data
4th data word, bits 17-24 of module data
5th data word, bits 1-16 of module data
6th data word, bits 17-24 of module data
4
data from the 1st channel
data from the 2nd channel
data from the 3rd channel
data from the 4th channel

Note that n^{th} channel has the channel number (n-1).

APPENDIX A: SUMMARY OF CAMAC COMMANDS

F(16)A(0)W(VSN)	write a 16-bit Virtual Station Number (VSN)
F(16)A(1)W(NumberOfModules)	write the number of modules to be serviced
F(16)A(2)W(ModuleData)	write the module data, one per module
F(16)A(3-6)W(DelaySubtr)	write the delay subtractors
F(16)A(7)W(GlobalMode)	write the operation mode data
F(16)A(8)W(GlobalDelays)	write the LAM timeout and trigger delay data
F(16)A(9,10)W(LAMMask)	write the LAM mask
F(16)A(11)W(ModuleDefinition)	define the user-defined type of module
F(0)A(0-10)	read back the data written with a F(16)A(0-10)
F(0)A(12)	read the firmware ID word
F(19)A(0-7, 12)W(PEROMData)	ATMEL AT29C010A PEROM operations
F(19)A(5)W(1)	Set reset for FPGA
F(19)A(5)W(2)	Release reset of FPGA (causes reboot).

NOTE: X=1 is generated in response to all valid commands.

APPENDIX B: OPERATIONS INVOLVING THE ATMEL AT29C010A PEROM

B.1. LOADING THE FPGA CONFIGURATION DATA INTO AT29C010A

The FPGA (XCS30 of the XILINX Spartan series) configuration file is 30996 bytes long. It is generated by a proper software (such as the XILINX Foundation Express) in a form of a binary or ASCII file. Both files start with some header data, which must be skipped so that only the true configuration data is downloaded into the ATMEL AT29C010A Programmable Erasable Read Only Memory (PEROM). The latter memory chip can accommodate up to four XCS30 configuration files. Which part of the chip is programmed or used by the FPGA (the active section) is decided by the setting of two jumpers J12 on the X-Port10 board. In the electronic schematics of X-Port10, these two jumpers set the values of the two most significant bits of the AT29C010A address.

No “erase” action is needed before reprogramming a PEROM. However, it should be verified that the software protection of the stored data is reset. After successful programming, it is recommended to set the software protection, as described in the next subsection.

For the duration of any PEROM operation (programming or software protection setting and resetting), the FPGA is forced to reset (its PRGM pin is pulled low). Upon completion of a PEROM operation, the FPGA will configure itself from the active section of the AT29C010A (selected by the two jumpers). A failure to configure is indicated by the yellow front-panel LED lighting up. This indicates, at the same time, a failure in programming the PEROM. In contrast, a successful operation is indicated by the green front-panel LED lighting up.

In the sample code below (Visual Basic), it is assumed that the configuration data reside in the first 30996 bytes of the array ConfigData(1 to 128*243) and that the subroutine to execute a Camac “write” command is CamacWrite16(N, A, F, W), where N, A, F, and W are the Camac N, A, F, and “write” data, respectively. The code reflects the fact that the AT29C010A PEROM requires bytes to be loaded by sectors of 128 bytes (243 sectors). This requires special actions (if...else...end if) to be taken at the sector boundaries. Also, because the length of the FPGA configuration file does not correspond to an integer number of sectors, the array ConfigData is longer than the configuration file (the unused bytes may be set to arbitrary values). In the code below, N points to the slot occupied by X-Port10. The function code for all PEROM operations is N = 19. Where the “write” data is arbitrary, it is indicated by an asterix, i.e., W = *.

‘code to load the FPGA configuration data into AT29C010A
dim ConfigData(1 to 31104) as byte

```
CamacWrite16(N, 4, 19, *)                    'enters the download mode
CamacWrite16(N, 7, 19, *)                    'resets the internal address counter
PointerToByte = 1
for SectorNumber = 1 to 243
        if SectorNumber = 1 then
                CamacWrite16(N, 7, 19, *)
        else
                CamacWrite16(N, 12, 19, *)
        end if
        for ByteInSector = 1 to 128
                if ByteInSector = 128 then
                        CamacWrite16(N, 6, 19, ConfigData(PointerToByte))
                else
                        CamacWrite16(N, 2, 19, ConfigData(PointerToByte))
                end if
                PointerToByte = PointerToByte + 1
        next ByteInSector
        Delay(20)
next SectorNumber
CamacWrite16(N, 5, 19, *)                    'exits the download mode
'end of the code
```

B.2. SOFTWARE DATA PROTECTION

Software data protection can be set and reset only when the 4th section of the AT29C010A is active (both jumpers of J12 on). The protection applies to all four sections.

The following sequence of seven Camac commands (the "write" data are in hex format) sets the protection:

```
'code to set the protection of the AT29010A data:
CamacWrite16(N, 4, 19, *)
CamacWrite16(N, 0, 19, *)
CamacWrite16(N, 1, 19, AAh)
CamacWrite16(N, 0, 19, 55h)
CamacWrite16(N, 3, 19, A0h)
CamacWrite16(N, 6, 19, 55h)
CamacWrite16(N, 5, 19, *)
'end of code
```

To reset the protection and, hence, enable reprogramming one must issue the following sequence of 10 CAMAC commands (the "write" data are in hex format):

```
'code to unprotect the AT29010A chip
CamacWrite16(N, 4, 19, *)
```

```
CamacWrite16(N, 0, 19, *)
CamacWrite16(N, 1, 19, AAh)
CamacWrite16(N, 0, 19, 55h)
CamacWrite16(N, 0, 19, 80h)
CamacWrite16(N, 1, 19, AAh)
CamacWrite16(N, 0, 19, 55h)
CamacWrite16(N, 3, 19, 20h)
CamacWrite16(N, 6, 19, AAh)
CamacWrite16(N, 5, 19, *)
‘end of code
```

APPENDIX C: DIAGNOSTIC MODE

The diagnostic mode of operating X-Port10 is an advanced feature intended for allowing one to monitor the timing of internal signals at the FIFO “Data In” and “Write” pins. This mode is useful in a fine-tuning of the timing of the “FIFO Write” signal, to speed up the CAMAC readout process. In this mode, the “FIFO Write” signal is routed to the ECL DWF (Diagnostic Write FIFO) port, parallel to the WST port, and a selected bit of the FIFO “Data In” signal is routed to the ECL DMX (Diagnostic Multiplexer) port, shared with the ECL PASS port. Additionally, a diagnostic trigger signal is routed to the ECL DTRG port.